Living Atlas 1 Sprint - 3 Report

Long, Joshua
Kolb, Mitchell, William
Svetlik, Sierra Amelia

Introduction

This sprint started on March 27th and ended on April 21st.. This sprint focused on combining our three codebases that we have worked on throughout the last two sprints into one singular working project.

Retrospection

Long, Joshua
During the previous sprint, our team made progress by dividing the Living Atlas project into separate layers such as Frontend, Backend, and Database. This allowed us to focus on our respective areas of expertise and make significant improvements. However, we encountered challenges when it came to integrating all three layers and creating a functional full-stack application. We also realized that we needed to clarify the type and structure of the data we were handling in order to ensure that our project was accurate and effective. Moving forward, we plan to collaborate more closely across the different layers and ensure that we have a clear understanding of the data being used in the Living Atlas application.

Kolb, Mitchell, William
Looking back at the last sprint I was tasked with to determine the language scheme for the requests of each endpoint in our backend, to connect the backend to the database to receive and enter information from it and to implement code that can demonstrate the four types of requests in restful API's like GET, POST, PUT, DELETE. I think these tasks were going to help move the project forward because they will allow the three of us to finally have all of our sections working synchronously. Overall this sprint was a success, I was able to help connect all three of our sections together through my backend as the middle step. One thing I wish I was able to do differently was to connect to the database first because when I coded the four requests I used hard coded data to learn how they work. I now realize I have to redo all of them to incorporate the database.

Svetlik, Sierra Amelia
During the second sprint, I was assigned the task of setting up the database. I had to create some of the tables according to the kind of information that the client wanted stored in the database, as well as have a way for the database to be accessible to the other members of the team. I decided on ElephantSQL as a remote database hosting service, which means that the

other team members can connect to it. I also created some tables for the database: a users table and a data table.

Result of Sprint - 3 Planning

Long, Joshua
One of the key tasks I have identified for Sprint-3 is creating an Upload form and ensuring that we have a clear understanding of the data we are handling. Additionally, we will enable GET and POST requests from the Frontend, allowing users to retrieve and submit data to the server. We will also focus on parsing the data into cards properly, ensuring that it is displayed in a user-friendly and visually appealing way. Ultimately, our goal is to integrate all three layers of the application seamlessly and create a fully functional data visualization feature for the Living Atlas project.

I picked these tasks for several reasons. First, creating an Upload form and understanding the data we are handling is crucial to the success of the Living Atlas project. We need to ensure that the data is structured correctly and can be easily uploaded and retrieved from the database. Integrating all three layers of the application is necessary for creating a fully functional product that meets the project goals. We are almost at the end of the semester and we need a working application.

Kolb, Mitchell, William
My task this sprint is to implement the 4 requests (GET, POST, PUT, DELETE) using the database connection that I established last sprint. I picked this task because this will use the skills I have acquired throughout this semester to get all the moving pieces together. I plan to create customized endpoints that will have a direct purpose in the front end so they can be predictably used and relied upon. I want to make the efficient and complete to the request at hand.

Svetlik, Sierra Amelia
An important task this sprint for me was to add a way to track categories, also known as tags, for data that will potentially be uploaded to the database. I also needed to put some fake data into the database so that the API could retrieve it and send it to the front end to be displayed. Adding the Tags table was important because potential users will search for data based on various categories. Putting fake data into the database was important as it would allow for testing the front end and API when the 3 sections were fully connected into a prototype of the application.

Sprint Task Assignment

Long, Joshua

1. FrontEnd - Design Upload button with POST request (issue #30 from Kanban board)
2. FrontEnd - cards must reflex GET Request (issue #32 from Kanban board)

I will be working on an upload form with the ability with POST request and a card component with the ability to send a GET request to the backend and fetch and parse the data.

Kolb, Mitchell, William

1. Backend - GET Request
2. Backend - DELETE Request
3. Backend - POST Request

I will be working on custom endpoints that incorporate the get, delete, and post functionality. I have decided to leave out the PUT request because the requires more knowledge of the database and the requirements of the purpose of the data that I do not have at the moment.

Svetlik, Sierra Amelia

1. Create the database

I will continue to work on refining the current database to match expectations from the client as well as what works with the API and front end. Challenges have come up from initial ideas for the database that did not work with the API and front end, as well as with expectations from the client, so the database is continuing to undergo modifications.

Sprint Task(s) Details

Long, Joshua

1. **FrontEnd - Design Upload button with POST request**

   An upload form that request all the necessary data from the user and able to send the form to the backend. I created a prototype of the Upload form and modify the design based on feedback from the project sponsor. Using React, I will create the user interface for the Upload form and use state management to store the input data. To handle uploads, I will use the Axios package to make a POST request to the backend.

2. **FrontEnd - cards must reflex GET Request**

   I plan to ensure that the Frontend cards in the Living Atlas application reflect the latest data retrieved from the server using a GET request. I will create a JavaScript function and use Axios to make the GET request to the server and parse the retrieved data into user-friendly cards using React. I will also use state management to store the retrieved data and ensure that it is properly formatted for display.

Kolb, Mitchell, William

1. **Backend - GET Request**

   To satisfy this task I will implement the get request in two endpoints called get_card() and get_all(). The card function will return the title and description for all the data points in our database to be displayed on the cards that are on the right side of the front end of the website. The all function will be used to return all the data of one particular card that is clicked on by the user.

2. **Backend - DELETE Request**

   To satisfy this task I will implement the delete_point() function which when given the title or the UID (user id) of a research point or user account will delete that row of information or all information relating to the user.

3. **Backend - POST Request**

   To satisfy this task I will implement the post_data() function which will take a form data structure that is passed to the back end from the front end. This form will contain the user submitted information that will need to be verified and parsed and then added to the database. This function will be more involved than the other one because I have to make sure the data is of the correct type and format for security reasons. This request will also take use of the INSERT query functionality which I have the least amount of experience in from my testing last sprint.

Svetlik, Sierra Amelia

1. **Create the database**

   To complete this task, I will continue to plan out the database as new information comes from the client and I learn more about the API and front end. I have sample data from the client that I can use to put in the database, and I will work closely with my team members to try and retrieve it.